

CLAIMS

We claim:

1. A process for monitoring, comprising:
5 accessing a method;
determining whether said method is complex; and
modifying said method for a particular purpose only if said method is complex.
2. A process according to claim 1, wherein:
10 said step of determining whether said method is complex includes determining
whether said method is non-synthetic.
3. A process according to claim 1, wherein:
said step of determining whether said method is complex includes determining
15 whether said method has an access level of public or package.
4. A process according to claim 1, wherein:
said step of determining whether said method is complex includes determining
whether said method calls another method.
20
5. A process according to claim 1, wherein:
said step of determining whether said method is complex includes determining
whether said method is non-synthetic, calls another method and has an access level of
public or package.
25
6. A process according to claim 1, wherein:
said step of determining whether said method is complex includes determining

whether said method calls one or more different methods and can be called by a sufficient scope of one or more other methods.

5 7. A process according to claim 1, wherein:
said step of modifying includes modifying object code.

8. A process according to claim 1, wherein:
said step of modifying includes adding a tracer for said method.

10 9. A process according to claim 1, wherein:
said step of modifying includes adding a timer for said method.

15 10. A process according to claim 1, wherein:
said step of modifying includes adding exit code and start code to existing object
code.

20 11. A process according to claim 10, wherein:
said start code starts a tracing process;
said exit code stops said tracing process;
said exit code is positioned to be executed subsequent to original object code;
said step of adding exit code includes adding an instruction to jump to said exit
code from said original object code;
said step of adding exit code includes adding an entry in an exceptions table; and
said step of adding an entry in said exceptions table includes adding a new entry
25 into said exceptions table for said method, said new entry indicates a range of indices
corresponding to said original object code, said new entry includes a reference to said
exit code and said new entry indicates that said new entry pertains to all types of

exceptions.

12. A process according to claim 1, wherein:
said particular purpose is to add a first tracer.

5

13. A process for monitoring, comprising:
determining which methods of a set of methods are complex; and
using a first tracing mechanism for said methods determined to be complex
without using said first tracing mechanism for methods not determined to be complex.

10

14. A process according to claim 13, wherein:
said step of determining includes determining whether said methods are non-
synthetic.

15

15. A process according to claim 12, wherein:
said step of determining includes determining whether said methods have an
access level of public or package.

20

16. A process according to claim 13, wherein:
said step of determining includes determining whether said methods call other
methods.

25

17. A process according to claim 13, wherein:
said step of determining includes determining whether said methods are non-
synthetic, call other methods and have an access level of public or package.

18. A process according to claim 13, wherein:

said step of determining includes determining whether said methods call one or more different methods and can be called by a sufficient scope of one or more other methods.

5 19. A process according to claim 13, wherein:

said step of using a first tracing mechanism includes adding and using timers for said methods.

 20. A process according to claim 13, wherein:

10 said step of using a first tracing mechanism includes modifying existing object code to add said first tracing mechanism.

 21. A process according to claim 20, wherein:

said first tracing mechanism includes timers for said methods.

15

 22. One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a process comprising:

determining which methods of a set of methods are complex; and

20 modifying for a particular purpose only those methods that are determined to be complex.

 23. One or more processor readable storage devices according to claim 22, wherein:

25 said step of determining includes determining whether said methods are non-synthetic.

24. One or more processor readable storage devices according to claim 22,
wherein:

said step of determining includes determining whether said methods have an
access level of public or package.

5

25. One or more processor readable storage devices according to claim 22,
wherein:

said step of determining includes determining whether said methods call other
methods.

10

26. One or more processor readable storage devices according to claim 22,
wherein:

said step of determining includes determining whether said methods are non-
synthetic, call other methods and have an access level of public or package.

15

27. One or more processor readable storage devices according to claim 22,
wherein:

said step of determining includes determining whether said methods calls one or
more different methods and can be called by a sufficient scope of one or more other
20 methods.

28. One or more processor readable storage devices according to claim 22,
wherein:

said step of modifying includes modifying existing object code.

25

29. One or more processor readable storage devices according to claim 22,
wherein:

said step of modifying includes adding tracers.

30. One or more processor readable storage devices according to claim 22,
wherein:

5 said step of modifying includes adding timers.

31. One or more processor readable storage devices according to claim 22,
wherein:

10 said step of modifying includes adding exit code and start code to existing object
code.

32. One or more processor readable storage devices according to claim 31,
wherein:

15 said start code starts a tracing process;
 said exit code stops said tracing process;
 said exit code is positioned to be executed subsequent to original object code;
 said step of adding exit code includes adding an instruction to jump to said exit
code from said original object code;
 said step of adding exit code includes adding an entry in an exceptions table; and
20 said step of adding an entry in said exceptions table includes adding a new entry
into said exceptions table for said method, said new entry indicates a range of indices
corresponding to said original object code, said new entry includes a reference to said
exit code and said new entry indicates that said new entry pertains to all types of
exceptions.

25

33. One or more processor readable storage devices having processor readable
code embodied on said processor readable storage devices, said processor readable code

for programming one or more processors to perform a process comprising:

determining whether a method is complex; and

tracing said method for a particular purpose only if said method is complex.

5 34. One or more processor readable storage devices according to claim 33,
wherein:

said step of determining whether said method is complex includes determining
whether said method is non-synthetic.

10 35. One or more processor readable storage devices according to claim 33,
wherein:

said step of determining whether said method is complex includes determining
whether said method has an access level of public or package.

15 36. One or more processor readable storage devices according to claim 33,
wherein:

said step of determining whether said method is complex includes determining
whether said method calls another method.

20 37. One or more processor readable storage devices according to claim 33,
wherein:

said step of determining whether said method is complex includes determining
whether said method is non-synthetic, calls another method and has an access level of
public or package.

25 38. One or more processor readable storage devices according to claim 3,
wherein:

said step of tracing includes timing said method.

39. An apparatus capable of monitoring, comprising:
means for determining whether a method is complex; and
5 means for tracing said method for a particular purpose only if said method is
complex.

40. An apparatus capable of monitoring, comprising:
a storage device; and
10 one or more processors in communication with said storage device, said one or
more processors perform a process comprising:
accessing a method,
determining whether said method calls one or more different methods and
can be called by a sufficient scope of one or more other methods, and
15 tracing said method for a particular purpose only if said method calls one
or more different methods and can be called by a sufficient scope of one or more other
methods.

41. An apparatus according to claim 40, wherein:
20 said step of determining includes determining whether said method is non-
synthetic.

42. An apparatus according to claim 40, wherein:
said step of determining includes determining whether said method has an access
25 level of public or package.

43. An apparatus according to claim 41, wherein:

said step of determining includes determining whether said method is non-synthetic.

- 5 44. An apparatus according to claim 40, wherein:
said process further includes modifying existing object code for said method in order to add a first tracing mechanism.

- 10 45. An apparatus according to claim 44, wherein:
said first tracing mechanism includes a timer.

46. An apparatus according to claim 40, wherein:
said step of tracing includes timing said method.